REMARKS/ARGUMENTS

Claims 1-6 and 9-14 are pending in the present application. Claims 7 and 8 are canceled. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 101

The Examiner rejected claims 7 and 8 as directed towards non-statutory subject matter. Applicants have canceled these claims, thereby rendering the rejection moot.

II. 35 U.S.C. § 102, Asserted Anticipation

The Examiner rejected claims 1, 3, 5, 7, and 14 under 35 U.S.C. § 102(e) as anticipated by *Schuba* et al., Network Protection for Denial of Service Attacks, U.S. Patent 6,725,378 (April 20, 2004) (hereinafter "*Schuba*"). This rejection is respectfully traversed. Regarding claim 1, the Examiner states that:

> **As to independent claim 1,** "A method of preventing a flooding attack on a network server" is taught in '378 col. 1, lines 55-60 "the present invention includes a unique defense for denial of service attacks";
>
> **"in which a large number of connectionless datagrams are received for queuing to a port on the network server, comprising:"** is shown in '378 col. 3, lines 16-33 "The Internet Protocol (IP) is the standard network layer protocol of the Internet that provides a connectionless, best effort packet delivery service. IP defines the basic unit of the data transfer used throughout an IP network, called a datagram. The deliver of datagrams is not guaranteed...Datagrams are routed towards their destination host" {"connectionless datagrams" same as "connectionless, best effort packet delivery service" / "network server" same as "destination host"};
>
> **"determining, in response to the arrival of a connectionles datagram from a host for a port on the network server"** is disclosed in '378 col. 4, lines 52-54 "When a SYN packet arrives at a port on which a TCP server is listening";
>
> **"if the number of connectionless; datagrams already queued to the port from the host exceeds a prescribed threshold discarding the datagram, if the number of connectionless datagrams already queued to the port from the host exceeds the prescribed threshold"** is taught in '378 col. 4, lines 54-58 "There is a limit on the number of concurrent TCP connections that can be in a half-open connection state, called the SYN-RECVD state (i.e., SYN received). When the maximum number of half-open connections per port is reached, TCP discards all new incoming connections requests";
>
> **"and queuing the connectionless datagram to a queue slot of the port, if the number of connectionless. datagrams already queued to the port from the host does not exceed the prescribed threshold"** is taught in '378 col. 4, lines

> 59-67 "until it has either cleared or completed some of the half-open connections"

Office Action dated May 2, 2006, pp. 5-6.

A prior art reference anticipates the claimed invention under 35 U.S.C. §102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case each and every feature of the presently claimed invention is not identically shown in the cited reference, arranged as they are in the claims.

Claim 1 is as follows:

> 1. A method of preventing a flooding attack on a network server in which a large number of connectionless datagrams are received for queuing to a port on the network server, comprising:
>
> determining, in response to the arrival of a connectionless datagram from a host for a port on the network server, if the number of connectionless datagrams already queued to the port from the host exceeds a prescribed threshold;
>
> discarding the datagram, if the number of connectionless datagram already queued to the port from the host exceeds the prescribed threshold; and
>
> queuing the connectionless datagram to a queue slot of the port, if the number of connectionless datagram already queued to the port from the host does not exceed the prescribed threshold.

Schuba does not anticipate claim 1 because *Schuba* does not teach the claimed steps of determining, discarding, and queuing, as claimed. *Schuba* does not teach these claimed steps because a half-open connection is not the same as a queue of datagrams. As proven in the previous response to office action, *Schuba* instead teaches discarding incoming *connection requests* until the maximum number of *half-open connections* is reduced. For example, *Schuba* states the following:

> *When a SYN packet arrives at a port on which a TCP server is listening, the above-mentioned data structures are allocated. There is a limit on the number of concurrent TCP connections that can be in a half-open connection state, called the SYN-RECVD state (i.e., SYN received). When the maximum number of half-open connections per port is reached, TCP discards all new incoming connection requests until it has either cleared or completed some of the half-open connections.* Typically, several ports can be flooded in this manner, resulting in degraded service or worse. Moreover, it should be appreciated that without a limit on the number of half-open connections, a different denial of service attack would result in which an attacker could request so many connections that the target machine's memory is completely exhausted by allocating data structures

for half-open TCP connections. Table II illustrates the half-open connection states that may be accommodated by various operating systems as follows:

TABLE II

Operating System	Backlog	Backlog + Grace
FreeBSD 2.1.5	n.a.	128
Linux 1.2.x	10	10
Solaris 2.4	5	n.a
Solaris 2.5.1	32	n.a.
SunOS 4.x	5	8
Windows NTs 3.51	6	6
Windows NTw 4.0	6	6

Schuba, col. 4, l. 52 through col. 5, l. 13 (emphasis to show portions cited by the Examiner).

Several important differences exist between discarding additional connection requests, as in *Schuba*, and discarding the datagram, if the number of connectionless datagrams already queued to the port from the host exceeds the prescribed threshold, as recited in claim 1. For example, in *Schuba*, no queue *for the datagrams themselves* has been described. Instead, *Schuba* refers to a *half-open backlog queue*, as shown by the following excerpt from *Schuba*:

> In accordance with these classifications, monitor 52 sends a RST packet to unacceptable or bad addresses, closing corresponding connections to free resources of destination hosts 54. Further, an ACK packet is sent for suspect source addresses to free resources of the destination hosts 54 by removing connections from ***a half-open backlog queue***. Further, states 108 of state machine 100, conditionally coupled by transition paths 112, 118, 124, 126, 134, and 136, permit suspect address classification to change based on observed behavior of network traffic and asynchronous events, such as expiry and staleness event timers.

Schuba, col. 11, ll. 16-26 (emphasis supplied).

In contrast, claim 1 requires "discarding the ***datagram***, if the number of connectionless datagrams already queued to the port from the host exceeds the prescribed threshold" (emphasis supplied). One of ordinary skill would instantly recognize the difference between discarding a datagram queued at a port and removing connections from a half-open backlog queue. In the previous response to office action, other distinctions between the two features were described. As an aid to understanding the distinction between this claimed feature and queues for half-open connections, the specification states:

> The first threshold is dynamically determined in the preferred embodiment. The owner of a server specifies for each port that is subject to datagram flooding checks a maximum number of queued datagrams (M) allowed at any time to the port and a controlling percentage (P) of available queue slots remaining for the port. The invention keeps track of the number (A) of queued datagrams for the port and it calculates the number of available queue slots (I) by subtracting the number of queued datagrams from the maximum number of datagrams (I = M –

> A). If the number of datagrams already queued for the transmitting host is equal to or greater than P times the number of queue slots left (=> P*I), then the present datagram is refused. Otherwise, the datagram is queued and the number of queued datagrams (A) for the port is incremented by one.

Specification, p. 3, l. 18 through p. 4, l. 6.

The specification and the claims unambiguously state that the queue at issue is for datagrams. A half-open connection is not a datagram, even if a half-open connection is created using datagrams. *Schuba* only teaches methods for dealing with too many half-open connections, which is entirely distinct from discarding datagrams queued at a port. The fact that half-open connections are created with connectionless datagrams is wholly irrelevant to this distinction. Therefore, *Schuba* does not anticipate claim 1.

Nevertheless, in the response to arguments the Examiner states that:

> In response to applicant's argument beginning on page 8, "Schuba does not anticipate claim 1 because Schuba does not teach the claimed steps of determining, discarding, and queuing, as claimed ... As shown below, a half-open connection is not the same as a queue of datagrams, contrary to any assumptions or assertions the examiner has made. Thus, Schuba does not teach "determining, in response to the arrival of a connectionless datagram from a host for a port on the network server". The Office disagrees with argument, Schuba does show teach discarding, and queuing as claimed, in viewing the arguments and lengthy case history with this application examiner finds applicant is trying to differentiate the meaning of a connection attempt by using word such as connectionless or queuing the connectionless datagram. The protection against flooding attack as claimed is shown in Schuba.
>
> In response to applicant's argument beginning on page 9, "As shown below, a half-open connection is not the same as a queue of datagrams, Schuba describes the process of establishing a transmission protocol (TCP) connection in figure 1 of Schuba ... As stated above, Schuba defines a half-open connection as a state in which the SYN datagram from a destination host has been received at a source host ... In contrast, the invention of claim 1 limits the number of datagrams that are allowed to queue at a given port". The Office disagree with argument and notes that "half open-connections" are interpreted to be equivalent to "queuing the connectionless datagram".

Office Action dated May 2, 2006, pp. 2-3.

In the first quoted paragraph, the Examiner states that "in viewing the arguments and lengthy case history with this application examiner finds applicant is trying to differentiate the meaning of a connection attempt by using word such as connectionless or queuing the connectionless datagram." However, the Examiner misconstrues Applicants' arguments. The thrust of Applicants' arguments is not directed towards splitting fine hairs over the meaning of the term "connectionless" or the meaning of the term "queuing the connectionless datagram." The thrust of Applicants' arguments is that a fundamental

and marked difference exists between a queue of connectionless datagrams at a port, as claimed, and a queue of half-open connections, as described in *Schuba*. As shown above and in the previous response to arguments, the two features are entirely distinct. Anyone of ordinary skill in the art understands the distinctions between these two features, though the Examiner steadfastly refuses to recognize this fact in the face of clear disclosure in *Schuba* to the contrary; additionally, the Examiner does not offer any support for the Examiner's assertions.

For example, in the second quoted paragraph, the Examiner states that, "The Office disagree with argument and notes that "half open-connections" are interpreted to be equivalent to "queuing the connectionless datagram"." The Examiner provided absolutely no support for the Examiner's assertions. The Examiner only quotes Applicants' argument and then asserts, without foundation or support, that "half open connections" are actually equivalent to "queuing the connectionless datagram." The Examiner provides no argument that the two are equivalent. Instead, the Examiner only misconstrued Applicants' argument and then concluded without any argument or support that the features are equivalent.

In view of the plain meaning of the claimed features and in view of the plain meaning of the teachings of *Schuba* quoted above, the Examiner's statements are incorrect. Applicants request that the Examiner either provide factual support for the assertion that a "half open connection" is equivalent to "queuing the connectionless datagram" or that the Examiner submit an affidavit under 37 C.F.R. § 104(d)(2) attesting to the Examiner's personal knowledge in this area. Thus, Applicants can have an adequate opportunity to consider and refute the basis for the Examiner's assertions.

Absent such support, the rejection and the Examiner's arguments are manifestly incorrect in view of the plain meaning of the claims and of *Schuba*. Instead, *Schuba* does not teach all of the features of claim 1, as described above. Therefore, *Schuba* does not anticipate claim 1.

Claims 3, 5, 7, and 14 all contain features similar to those presented in claim 1. Therefore, *Schuba* does not anticipate these claims for the reasons presented above.

Furthermore, *Schuba* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Absent the Examiner pointing out some teaching or incentive to implement *Schuba* and discarding queued connectionless datagrams as claimed, one of ordinary skill in the art would not be led to modify *Schuba* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *Schuba* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

III. 35 U.S.C. § 103, Asserted Obviousness

The Examiner rejected claims 2, 4, 6, and 8-13 under 35 U.S.C. §103 as obvious over *Schuba* in

view of *Yavatkar* et al., Method and System for Diagnosing Network Intrusion, U.S. Patent 6,735,702 (May 11, 2004) (hereinafter "*Yavatkar*"). This rejection is respectfully traversed. Regarding claim 2, the Examiner states that:

> As to dependent claim 2, the following is not taught in '378 "wherein the determining if the number of datagrams already queued to the port from the host exceeds a prescribed threshold further comprises: calculating the prescribed threshold by multiplying a percentage by the number of available queue slots for the port" however '702 teaches "A watchdog agent may assume a network attack exist if network congestion is detected...In an alternate embodiment a watchdog agent detects network congestion by monitoring interface discard counts and average queue lengths for each port on the node" in col. 15, line 63 through col. 16, line 17.
>
> It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of '378 a method to protect a network from denial of service attacks to include a means to calculate the threshold limit per port. One of ordinary skill in the art would have been motivated to perform such a modification in order to gain information needed to diagnose a network attack (see '702 col. 2 lines 44 et seq.) "Therefore there exists a need for a system and method allowing for the distributed state of a network such as information about attack traffic, to be quickly and accurately collected. A system and method are needed for quickly and accurately diagnosing network attacks by determining information such as the source of, or a partial path of, attack traffic".

Office Action dated May 2, 2006, p. 7.

If the Patent Office does not produce a *prima facie* case of unpatentability, then without more the applicant is entitled to grant of a patent. *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); *In re Grabiak*, 769 F.2d 729, 733, 226 U.S.P.Q. 870, 873 (Fed. Cir. 1985). A *prima facie* case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). In this case, the teachings of the prior art doe not suggest the claimed subject matter to a person of ordinary skill in the art.

The Examiner has failed to state a *prima facie* obviousness rejection because the proposed combination does not teach all of the features of the claims. Claims 4, 6, and 8-13 depend from independent claims 1, 3, 5, or 7. As shown above, *Schuba* does not teach all of the features of the independent claims. Furthermore, *Schuba* does not suggest the features of the independent claims because *Schuba* describes an entirely different method of dealing with flooding attacks compared with the method of claim 1 and with the features of the other independent claims. The Examiner tacitly admits that *Yavatkar* does not teach the features of the independent claims because the Examiner does not assert

otherwise and because the Examiner would not otherwise need to rely on *Schuba*. Thus, the proposed combination does not teach or suggest all of the features of the independent claims. Accordingly, the Examiner has failed to state a *prima facie* obviousness rejection against claims 4, 6, and 8-13 at least by virtue of their dependence on the respective independent claims.

In addition, the proposed combination does not teach all of the features of the other dependent claims. For example, neither *Yavatkar* nor *Schuba* teach or suggest the feature of "configuring a maximum number of connectionless datagrams allowed to be queued at the port," as claimed in claim 9. The Examiner tacitly admits that *Schuba* does not teach this claimed feature. The Examiner states that *Yavatkar* does teach this claimed feature as follows:

> **As to dependent claim 9, "further comprising: configuring a maximum number of connectionless, datagrams allowed to be queued at the port"** is taught in '702 col, 12, lines 27-39 "In step 440, proactive environment 100 instantiates service object 300 based on the class of service 102. Proactive environment 100 configures service object 300 per the permissioning accessed in step 434. For example, one set of permissioning may allow agent 110 to use service object 300 to read the operating characteristics of port 21 and alter settings for the port".

Office Action of November 14, 2005, p. 6 (emphasis in original).

The portion of *Yavatkar* cited by the examiner is as follows:

> In step 440, proactive environment 100 instantiates service object 300 based on the class of service 102. Proactive environment 100 configures service object 300 per the permissioning accessed in step 434. For example, one set of permissioning may allow agent 110 to use service object 300 to read the operating characteristics of port 21 and alter settings for the port, and another set of permissioning may allow agent 110 to use service object 300 only to read the operating characteristics of port 21. Proactive environment 100 sets permission variables 312, members of service object 300, to indicate which aspects of service 102 (in the form of methods 322-326 of service object 300) agent 110 may access.

Yavatkar, col. 12, ll. 27-39.

The cited text plainly does not teach or suggest "configuring a maximum number of connectionless datagrams allowed to be queued at the port," as claimed in claim 9. Nothing else in *Yavatkar* teaches or suggests this claimed feature. Because neither *Schuba* nor *Yavatkar* teach or suggest this claimed feature, the proposed combination when considered as a whole does not teach or suggest this claimed feature. Therefore, the Examiner has failed to state a *prima facie* obviousness rejection of claim 9.

In summary, the Examiner has failed to state a *prima facie* obviousness rejection against any of the claims because the proposed combination when considered as a whole does not teach or suggest all of the features of the claims. Accordingly, the rejection of claims 2, 4, 6, and 8-13 has been overcome.

IV. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: August 2, 2006

Respectfully submitted,

/Theodore D. Fay III/

Theodore D. Fay III
Reg. No. 48,504
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants